CHAPTER 4 Secure Design Principles

Every network security implementation is based on some kind of model, whether clearly stated as such or assumed. For example, organizations that use firewalls as their primary means of defense rely on a perimeter security model, while organizations that rely on several different security mechanisms are practicing a layered defense model. Every security design includes certain assumptions about what is trusted and what is not trusted, and who can go where. Starting out with clear definitions of what is fully trusted, what is partially trusted, and what is untrusted, along with an understanding of which defense model is being used, can make a security infrastructure more effective and applicable to the environment it is meant to protect.

The CIA Triad and Other Models

Every security book written in the last several years mentions the CIA triad—Confidentiality, Integrity, and Availability. This venerable, well-established conceptual model, though very data-centric, is often useful in helping people think about security in terms of the most important aspects of information protection.

The CIA concept is not perfect. CIA focuses on three aspects of information protection that indeed are important, but it is not an all-inclusive model. Throughout this book, you will find many more important concepts in addition to these three, but they are mentioned here for the sake of completeness and consistency with common vocabulary. You should keep in mind that not all security professionals are big fans of the CIA triad, but you should be familiar with it.

Confidentiality

Confidentiality refers to the restriction of access to data only to those who are authorized to use it. Generally speaking, this means a single set of data is accessible to one or more authorized people or systems, and nobody else can see it. Confidentiality is distinguishable from *privacy* in the sense that "confidential" implies access to one set of data by many sources, while "private" usually means the data is accessible only to a single source. As an example, a password is considered private because only one person should know it, while a patient record is considered confidential because multiple members of the patient's medical staff are allowed to see it. Confidentiality controls are described in the chapters contained in Part II of this book.

Integrity

Integrity, which is particularly relevant to data, refers to the assurance that the data has not been altered in an unauthorized way. Integrity controls are meant to ensure that a set of data can't be modified (or deleted entirely) by an unauthorized party. Part of the goal of integrity controls is to block the ability of unauthorized people to make changes to data, and another part is to provide a means of restoring data back to a known good state (as in backups). Data integrity is also covered in the chapters of Part II of this book (from a design perspective) and Part IV (from an operations perspective).

Availability

Unlike confidentiality and integrity, which make the most sense in the context of the data contained within computer systems, availability refers to the "uptime" of computer-based services—the assurance that the service will be available when it's needed. Service availability is usually protected by implementing high-availability (or continuous-service) controls on computers, networks, and storage. High-availability (HA) pairs or clusters of computers, redundant network links, and RAID disks are examples of mechanisms to protect availability.

Additional Concepts

Alternatives to the CIA triad that include other aspects of security have been proposed by various thought leaders in the security profession. For example, Donn B. Parker proposed a set of six elements, known as the Parkerian Hexad, or the six atomic elements of information, which includes Control (or Physical Possession), Authenticity, and Utility. Other principles that have been proposed include Accountability, Non-Repudiation, and Legality. The U.S. Department of Defense defined "Five Pillars of Information Assurance," which include Authenticity and Non-Repudiation along with the CIA triad. The Organization for Economic Co-operation and Development (OECD) published guidelines that added Awareness, Responsibility, Response, Ethics, Democracy, Risk Assessment, Security Design and Implementation, Security Management, and Reassessment. Perhaps the most complete set is included in the U.S. National Institute of Standards and Technology Special Publication 800-27, Revision A, which proposes a total of 33 principles for securing technology systems. All of the various concepts used to break security into logical categories are included throughout this book. As you can see, there are many ways to categorize security principles, and the CIA triad is the most simplistic of them all.

In sum, the best-known attributes of security defined in the preceding models and others like them include

- Confidentiality
- Integrity
- Availability
- Accountability
- Accuracy
- Authenticity
- Awareness
- Completeness
- Consistency
- Control
- Democracy
- Ethics
- Legality
- Non-repudiation
- Ownership
- Physical Possession
- Reassessment
- Relevance
- Response
- Responsibility
- Risk Assessment
- Security Design and Implementation
- Security Management
- Timeliness
- Utility

Defense Models

Getting back to basics—what's the best way to defend against threats to the assets you want to protect? There are two approaches you can take to preserve the confidentiality, integrity, availability, and authenticity of electronic and physical assets such as the data on your network:

- Build a defensive perimeter around those assets and trust everyone who has access inside
- Use many different types and levels of security controls in a layered defense-in-depth approach

The first edition of this book introduced the concepts of the lollipop and the onion to visually depict the two most common approaches to security.

The Lollipop Model

The most common form of defense, known as *perimeter security*, involves building a virtual (or physical) wall around objects of value. Perimeter security is like a lollipop with a hard, crunchy shell on the outside and a soft, chewy center on the inside, as illustrated in Figure 4-1. Consider the example of a house—it has walls, doors, and windows to protect what's inside (a perimeter). But does that make it impenetrable? No, because a determined attacker can find a way in—either by breaking through the perimeter, or exploiting some weakness in it, or convincing someone inside to let them in. By comparison, in network security, a firewall is like the house—it is a perimeter that can't keep out all attackers. Yet the firewall is the most common choice for controlling outside access to the internal network, creating a

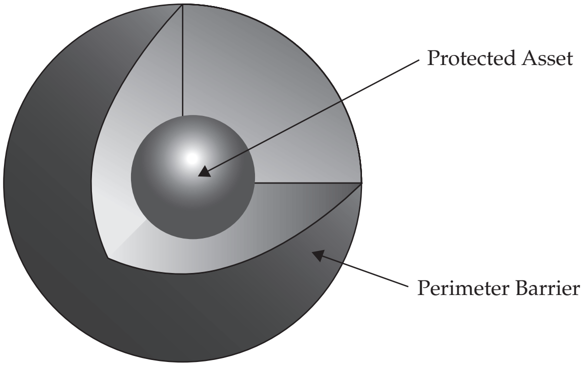

Figure 4-1 The lollipop model of defense

virtual perimeter around the internal network (which is usually left wide open). This often creates a false sense of security, because attackers can break through, exploit vulnerabilities, or compromise the network from the inside.

One of the limitations of perimeter security is that once an attacker breaches the perimeter defense, the valuables inside are completely exposed. As with a lollipop, once the hard, crunchy exterior is cracked, the soft, chewy center is exposed. That's why this is not the best model of defense.

Another limitation of the lollipop model is that it does not provide different levels of security. In a house, for example, there may be jewels, stereo equipment, and cash. These are all provided the same level of protection by the outside walls, but they often require different levels of protection. On a computer network, a firewall is likewise limited in its abilities, and it shouldn't be expected to be the only line of defense against intrusion.

NOTE A lollipop defense is not enough to provide sufficient protection. It fails to address inside threats and provides no protection against a perimeter breach. Yet many organizations do not understand firewalls in this way. Firewalls are an important part of a complete network security strategy, but they are not the only part. A layered approach is best.

Firewalls are an important part of a comprehensive network security strategy, but they are not sufficient alone. Today, networks both send information to and receive information from the Internet, and the rules for doing so are complex. Firewalls are still useful for shielding networks from each other, but they are often not sufficient to provide proper access controls, especially when internetwork communication and network resource sharing are complicated.

The Onion Model

A better approach is the *onion model* of security. It is a layered strategy, often referred to as *defense in depth*. This model addresses the contingency of a perimeter security breach occurring. It includes the strong wall of the lollipop but goes beyond the idea of a simple barrier, as depicted in Figure 4-2. A layered security architecture, like an onion, must be peeled away by the attacker, layer by layer, with plenty of crying.

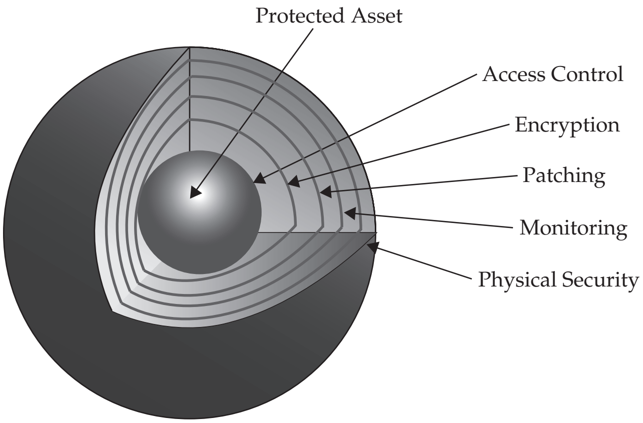

Figure 4-2 The onion model of defense

Consider what happens when an invader picks the front door lock or breaks a window to gain entry to a house. The homeowner may hide cash in a drawer and may store valuable jewels in a safe. These protective mechanisms address the contingency that the perimeter security fails. They also address the prospect of an inside job. The same principles apply to network security. What happens when an attacker gets past the firewall? What happens when a trusted insider, like an employee or a contractor, abuses their privileges? The onion model addresses these contingencies.

A firewall alone provides only one layer of protection against threats originating from the Internet, and it does not address internal security needs. With only one layer of protection, which is common on networks connected to the Internet, all a determined individual has to do is successfully attack that one system to gain full access to everything on the network. A layered security architecture provides multiple levels of protection against internal and external threats.

The more layers of controls that exist, the better the protection against a failure of any one of those layers. Consider a system that allows full access to an account that only uses username/password authentication, without any other security controls. That system uses only one layer of security, and it is strictly an authentication control. Anyone who obtains the username and password, or hijacks an account that's already logged in, can gain full access to the system. Since there are no other layers that must be bypassed, the system would be completely compromised. If such a system had further layers of security controls that needed to be passed after the username and password authentication, compromising the system would be correspondingly more difficult.

The layered security approach can be applied at any level where security controls are placed, not only to increase the amount of work required for an attacker to break down the defenses, but also to reduce the risk of unintended failure of any single technology. System, network, and application authentication controls can be layered. Network and system access controls can also be layered. Encryption protocols can be layered (such as by encrypting first with PGP followed by encrypting with Blowfish or AES). Audit trails can be layered with the use of local system logs coupled with off-system network activity logs.

Merging Security Models: A Case Study

Two well-known computer manufacturers, Silicon Graphics (SGI) and Cray Research, merged in an attempt to combine SGI's three-dimensional modeling and display technologies with Cray's supercomputing technologies. Brent Chapman, a well-known information security professional and author, gave a presentation about this merger and the resulting security architecture while the merger was under development. The merger of these two organizations presented a unique and interesting challenge to security architects.

SGI had an open, casual, collaborative group of technical engineers who enjoyed full, unrestricted access. Their computer network was based on the lollipop defense model. Everything on the inside was freely accessible by every employee. Cray, in contrast, had a cautious, clearly defined set of duties for every job position. Its network was highly segmented and followed the onion defense model. These employees were required to demonstrate a need to know before they were given access to areas of the network. After the business end of the merger was completed, both organizations were faced with a difficult decision about how to connect their two networks to fit their corporate cultures.

Because the two organizations had very different corporate cultures, they required different security models for different parts of their networks. The solution was called *containment fields*, which used firewalls as access control mechanisms to segment networks with differing security requirements. For example, there were classrooms and demonstration facilities that customers were allowed access to, as well as internal development networks where outsiders were strictly prohibited. Containment fields were developed as a way to establish and link pieces of networks that had special security requirements without compromising the security of the larger network in which they resided.

System availability controls can be layered by using clustering technology and redundancy. Many organizations use uninterruptible power supply (UPS) systems but also have backup generators in case the UPS systems fail. These are all examples of layered approaches that place similar controls in conjunction, or in sequence, to compensate for the loss of any individual control.

Zones of Trust

Different areas of a network trust each other in different ways. Some communications are trusted completely—the services they rely on assume that the sender and recipient are on the same level, as if they were running on a single system. Some are trusted incompletely—they involve less trusted networks and systems, so communications should be filtered. Some networks (like the Internet or wireless hot spots) are untrusted. The security controls should carefully screen the interfaces between each of these networks. These definitions of trust levels of networks and computer systems are known as *zones of trust*. This concept is illustrated in Figure 4-3.

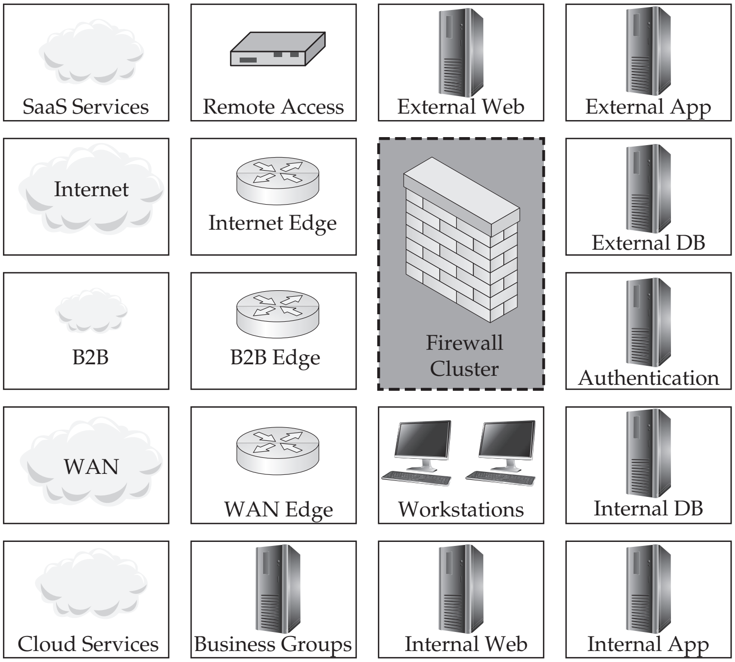

Figure 4-3 Zones of trust

Once you have identified the risks and threats to your business, and you know what functions are required for your business, you can begin to separate those functions into zones of trust. To do this, you need to assign levels of trust to each collection of resources on the network—in other words, you need to specify what level of risk is acceptable to accomplish each business function. That involves making trade-offs between what you want to do and what you want to avoid.

Zones of trust are connected with one another, and business requirements evolve and require communications between various disparate networks, systems, and other entities on the networks. Corporate mergers and acquisitions, as well as business partner relationships, produce additional complexities within the networking environment that can be diagrammed and viewed from the perspective of trust relationships. Once you understand how systems need to communicate with each other on the network, you can begin to develop a strategy for containing those systems into zones.

NOTE Different levels of trust are always present in any environment. Some areas are trusted more than others, and different areas trust each other in different ways. Enumerating these areas is an important step in reducing the weak spots that can undermine a security implementation.

IT resources vary in the extent to which they trust each other. Separating these resources into zones of trust enables you to vary the levels of security for these resources according to their individual security needs. The use of multiple zones allows access between a less and a more trusted zone to be controlled to protect a more trusted resource from attack by a less trusted one. Any zone could be subdivided into *policy pockets* of common security policies if need be, to support additional classification categories without the infrastructure expense of establishing another zone.

To visualize trust zones, imagine a castle surrounded by multiple walls that form concentric rings around the castle. There are cities in the rings, and there is exactly one door in each of the ring walls. Each door has a guard who says "Who goes there?" and who may ask for identification and a password. It is difficult for people in outer rings to attack people in the inner rings, but it isn't difficult to attack people if they are in the same ring. Thus, those in the same ring need to have the same minimum level of trustworthiness.

To establish a minimum level of trust, each zone (except perhaps an "untrusted" zone) requires that the devices in it have a certain, equivalent level of security—this level of security is determined by the technologies and procedures that are in place to check for attacks, intrusions, and security policy violations. Measures to establish trust include fixing known problems, detecting intrusions, and periodically checking for unauthorized changes, violations of policy, and vulnerabilities to attack.

Firewalls, routers, virtual LANs (VLANs), and other network access control devices and technologies can be used to separate trust zones from each other (as the walls in the castle analogy did). Access control lists (ACLs) and firewall rules can be used to control the intercommunication between these levels, based on authorization rules defined in the security architecture.

The importance of trust models is that they allow a broad, enterprise-wide view of networks, systems, and data communications, and they highlight the interactions among all of these components. Trust models can also distinguish boundaries between networks and systems, and they can identify interactions that might otherwise be overlooked at the network level or system level.

Trust can also be viewed from a transaction perspective. During a particular transaction, several systems may communicate through various zones of trust. In a transaction-level trust model, instead of systems being separated into different trust zones based on their locations on the network (as is done with the Internet, a DMZ, and an internal network), systems can be separated into functional categories based on the types of transactions they process. For example, a credit card transaction may pass through a web server, an application server, a database, and a credit-checking service on the Internet. During the transaction, all of these systems must trust each other equally, even though the transaction may cross several network boundaries. Thus, security controls at the system and network levels should allow each of these systems to perform their authorized functions while preventing other systems not involved in the transaction from accessing these resources.

Segmenting network data resources based on their access requirements is a good security practice. Segmentation allows greater refinement of access control based on the audience for each particular system, and it helps confine the communications between systems to the services that have transactional trust relationships. Segmentation also confines the damage of a security compromise. In the event that a particular system is compromised, network segmentation with access control lists reduces the number and types of attacks that can be launched from the compromised system. For example, web servers often experience compromises due to the ease and flexibility of web server attacks. A compromised web server that is confined in its own network segment offers fewer opportunities for the attacker to continue attempting to attack other servers.

NOTE Network segmentation is an important component of network security because it contains the damage caused by network intrusions, malfunctions, and accidents.

A layered segmentation approach also provides a useful conceptual model for network and system administrators. Several groups of servers can be included in a layer, defined by the types of services they perform, the types of data they handle, and the places they need to communicate to and from. For example, a public layer may contain systems that accept communication directly from the Internet. An application layer may contain systems that accept communication from the public layer. A data layer may accept communication from the application layer. Communication between these layers can be managed by a firewall, or by ACLs.

Best Practices for Network Defense

It only takes one careless end user to infect an entire network. If you are an administrator, it is clear that all the good intentions and friendly newsletters will not assure a reasonable level of computer security. You must stop malicious mobile code from arriving on the desktop in the first place, close holes, and make sure the users' computers are appropriately configured. If they can't click on malware, run it, or allow it on their computer, you've significantly decreased the threat of malicious attack.

There are many countermeasures you can implement to minimize the risk of a successful attack, such as securing the physical environment, hardening the operating systems, keeping patches updated, using an antivirus scanner, using a firewall, securing network share permissions, using encryptions, securing applications, backing up the system, creating a computer security defense plan, and implementing ARP poisoning defenses.

Secure the Physical Environment

A basic part of any computer security plan is the physical aspect. Of course, mission-critical servers should be protected behind a locked door, but regular PCs need physical protection too. Depending on your environment, PCs and laptops might need to be physically secured to their desks. There are several different kinds of lockdown devices, from thin lanyards of rubber-coated wire to hardened metal jackets custom-made to surround a PC. If anyone leaves their laptop on their desk overnight, it should be secured. There are also other steps that need to be taken on every PC in your environment.

Password Protect Booting

Consider requiring a boot-up password before the operating system will load. This can usually be set in the CMOS/BIOS and is called a user or boot password. This is especially important for portable computers, such as laptops and tablets and smartphones. Small-form-factor PCs are the most likely candidates to be stolen. Since most portable devices often contain personal or confidential information, password-protecting the boot sequence might keep a nontechnical thief from easily seeing the data on the hard drive or storage RAM. If a boot-up password is reset on a tablet or smartphone, often it requires that the data be erased too, so confidentiality and privacy are assured.

Password Protect CMOS

The CMOS/BIOS settings of a computer contain many potential security settings, such as boot order, remote wake-up, and antivirus boot-sector protection. It is important to ensure that unauthorized users do not have access to the CMOS/BIOS settings. Most CMOS/BIOSs allow you to set up a password to prevent unauthorized changes. The password should not be the same as other administrative passwords, but for simplicity's sake, a common password can be used for all machines.

There are ways around the CMOS/BIOS and boot-up passwords. Some boot-up passwords are able to be bypassed by using a special bootable floppy disk from the motherboard manufacturer or by changing a jumper setting on the motherboard. While they are not 100 percent reliable, a CMOS/BIOS or boot-up password might prevent some attacks from happening. For instance, in the Chapter 2 example where the gray hat attacker (a trusted conference presenter) oversupplied the security guard with soda, his physical attack was successful because he was able to slip into the unguarded room, put a floppy disk in the drive, and reboot the server on it. Had the floppy disk drive been disabled in the CMOS/BIOS and the boot sequence password-protected, his attack probably would have been unsuccessful.

Various operating system boot loaders, like Linux's LILO, allow boot-up passwords to be set. Of course, that won't stop someone from booting from another drive with a similar file system and taking over the machine. That's why the next step is so important.

Disable Booting from USB and CD

Disabling booting from USB storage devices and optical drives will prevent boot viruses from those devices and stop attackers from bypassing operating system security by loading a different operating system on the computer.

Harden the Operating System

As described in Chapters 21 and 22, reduce the *attack surface* of the operating system by removing unnecessary software, disabling unneeded services, and locking down access:

1. Reduce the attack surface of systems by turning off unneeded services.
2. Install secure software.
3. Configure software settings securely.
4. Patch systems regularly and quickly.
5. Segment the network into zones of trust and place systems into those zones based on their communication needs and Internet exposure.
6. Strengthen authentication processes.
7. Limit the number (and privileges) of administrators.

Keep Patches Updated

An attacker's best friend is an unpatched system. In most cases, the vulnerabilities used are widely known, and the affected vendors have already released patches for system administrators to apply. Unfortunately, a large percentage of the world does not regularly apply patches, and attacks against unpatched systems are widely successful. A solid patch-management plan is essential for protecting any platform, regardless of operating system and regardless of whether or not it is connected directly to the Internet.

Chapter 21 provides some best practices for keeping Unix system software updated with security updates, and Chapter 22 gives advice on Windows patch management. Operating system security updates are also important for network devices (Chapter 14), mobile devices (Chapter 25), and other infrastructure (Chapter 23). Basically, keeping any technology system up-to-date with the latest software is crucial, because vendors find and fix vulnerabilities over time. Don't let a vulnerability hang around on your systems waiting for an attacker to exploit.

Use an Antivirus Scanner (with Real-Time Scanning)

In today's world, an antivirus (AV) scanner is essential. It should be deployed on your desktop, with forced, automatic updates, and it should be enabled for real-time protection. Although deploying an AV scanner on your e-mail gateway is a good secondary or adjunct choice, if you only have the money to deploy an AV scanner in one location, choose the desktop. Why? Because no matter how the malware comes in (whether by e-mail, storage device, wireless, macro, Internet, tablet/smartphone, P2P, or IM), it must execute on the desktop to start harming. By placing the antivirus solution on the desktop, you are ensuring that no matter how it gets there, it will be blocked. E-mail and gateway AV-only solutions work most of the time, but they will fail if the malware comes in via any other method or on an unexpected port.

The AV solution should be enabled for real-time protection so it scans every file as it comes into the system or enters the computer's memory, so it can prevent malware from executing. Sometimes, in the interest of performance, users will want to disable the real-time functionality. Avoid agreeing to these requests, because that real-time scanning, even if it affects performance, is your best protection against infection.

Use Firewall Software

Almost as important as an AV scanner is the firewall. Firewalls have come a long way since their days of simple port filtering. Today's devices are stateful inspection systems capable of analyzing threats occurring anywhere in layers three through seven with software that runs directly on the computer. Firewalls are able to collate separate events into one threat description (such as a port scan) and can identify the attack by name (such as a teardrop fragmentation attack). Every PC should be protected by firewall software.

Desktop firewall software (also known as host-based firewalls or personal firewall software) can protect a PC against internal and external threats and usually offer the added advantage of blocking unauthorized software applications (such as Trojans) from initiating outbound traffic. Many antivirus scanning organizations offer firewall combo packages.

Secure Network Share Permissions

One of the most common ways an attacker or worm breaks into a system is through a network share (such as NetBIOS or SMB) with no password or a weak password. Folders and files accessed remotely over the network should have discretionary ACLs (DACLs) applied using the principle of least privilege and should have complex passwords.

By default, Windows assigns, and most administrators allow, the Everyone group to have Full Control or Read permissions throughout the operating system and on every newly created share. This is the opposite of the least privilege principle (maybe it should be called the most privilege principle). To counteract this problem, you should, at a minimum, begin by changing Everyone Full Control to Authenticated Users Full Control, wherever you can. Although this is not really that much more secure than the former setting, it will stop unauthenticated users, like anonymous and Guest users, from getting Full Control of resources by default.

Many Windows administrators also believe that it is acceptable for all shares to have Everyone Full Control because the underlying NTFS permissions, which are usually less permissive, will result in the desired tighter effective permissions. While this is true if you are 100 percent accurate in setting NTFS permissions, it goes against the principle of defense in depth (the onion model). A better strategy is to assign share and NTFS permissions to the smallest allowable list of groups and users. That way, if you accidentally set your NTFS file permissions too open, the share permissions might counteract the mistake.

Use Encryption

Most computer systems have many encryption opportunities. Use them. Linux and Unix administrators should be using SSH instead of Telnet or FTP to manage their computers. The latter utilities work in plaintext over the network, whereas SSH is encrypted. If you must use FTP, consider using an FTP service that uses SSL and digital certificates to encrypt traffic. In order for encrypted FTP to work, both the client and the server must support the same encryption mechanism. Use Windows IPSec Policies to require encrypted communications between servers and clients.

Encrypting File System (EFS) is one of the most exciting features in Windows. EFS encrypts and decrypts protected files and folders on the fly. Once turned on by a user, EFS will automatically generate public/private encryption key pairs for the user and the recovery agent. All the encrypting and decrypting is done invisibly in the background. If an unauthorized user tries to access an EFS-protected file, they will be denied access.

TIP To turn on EFS, right-click a file or folder, choose the Properties tab, click the Advanced button under the Attributes section, and then choose Encrypt Contents to Secure Data.

Because EFS encrypts and decrypts on the fly, it won't prevent malware occurrences while the authorized user is logged on. However, EFS-protected folders and files will be protected when the authorized user is not logged on. This may prevent maliciousness in certain circumstances, like a widespread worm attack that is running amok on a file server, corrupting every data file it can find (like the VBS.Newlove worm does). Since EFS can help provide additional security, is virtually invisible to the end user, and has a minimal performance hit, it is something to consider using for added protection.

Secure Applications

Managing your applications and their security should be a top priority of any administrator. Applications can be managed by configuring application security, installing applications to nonstandard directories and ports, locking down applications, securing P2P services, and making sure your application programmers code securely.

Securely Configure Applications

Applications should be configured with the vendors' recommended security settings. The three most commonly exploited Windows applications are Microsoft's Outlook (Express), Internet Explorer, and the Microsoft Office suite of applications. These applications may belong on end user workstations where people need them to do work, but they probably don't belong on your organization's servers. If you need high security on your servers, remove these applications. Because of the risk of common exploits, servers should not have e-mail clients (e.g. Outlook) or Microsoft Office installed on them.

In end-user PC environments, however, you want to keep the applications and minimize the risk at the same time. You can do this by regularly applying security patches and making sure security settings are set at the vendor's recommended settings, if not higher. Outlook and Outlook Express should both have their security zone set to Restricted. Internet Explorer's Internet zone should be set to Medium-High or High. Office offers administrative templates (called ADM files) that can be configured and deployed using System Policies or Group Policies. These can be downloaded from Microsoft's web site or found on the Office Resource Kit.

Other applications usually come with default security settings, and you can visit the vendor's technical support resources to find out more about your security choices. Unfortunately, many software vendors don't take security seriously. That's when you will need to use the concepts and practices you've learned from this book, and you may need to do some research on your own. If an exploit becomes known that targets your application, it usually shows up on the common security websites and mailing lists. One of the most inclusive exploit notification newsletters can be found at SANS (www.sans.org). SANS publishes weekly lists of all exploits affecting almost any operating system platform, including Windows, Unix, Linux, Macintosh, FreeBSD, and more.

Securing E-Mail E-mail worms continue to be the number-one threat on computer systems, especially Windows systems running Outlook or Outlook Express. Most worms arrive as a file attachment or as an embedded script that the end user executes. Clearly, you can significantly decrease your network's exposure risk by securing e-mail. This can be done by disabling HTML content and blocking potentially malicious file attachments.

Anything beyond plain text in an e-mail can be used maliciously against a computer. For that reason, it is important to restrict e-mails to plain text only or, if you must allow it, plain HTML coding only. You should disable scripting languages and active content, such as ActiveX controls, Java, and VBScript objects. Often this is as simple as checking a checkbox in the e-mail client to force all incoming e-mail to be rendered in plain text. Some clients handle this more elegantly than others, and HTML-only messages can be badly mangled during conversion or can appear blank. Outlook and Outlook Express allow e-mails with active content to be opened in the Restricted Internet zone, which disables content beyond plain HTML coding. This is the default setting in Microsoft's latest e-mail clients. Early clients opened e-mail in the much more permissive Internet security zone.

If you can block active content from executing, then all you have to worry about is end users clicking malicious HTML links or opening file attachments. It is difficult to block users from clicking malicious HTML links if they already have Internet access. In Windows environments, you can use Group Policy, Internet Explorer Administration Kit (IEAK), or some other type of proxy server filter to only allow end users to visit preapproved sites, but beyond that you have to rely on end-user education.

Blocking Dangerous File Types Blocking dangerous file attachments is the best way to prevent exploits, given today's preferred method of e-mailing viruses and worms. The biggest question is "What constitutes a dangerous file type?" The truth is that almost any file type can be used maliciously, so the better question is "What are the popularly used malicious file types?" Even that list isn't small. Table 4-1 shows the Windows file types that are commonly blocked in organizations that are concerned about the various popular attacks that use these file types as vectors. These are in order of their prevalence in e-mail server block lists.

File Extension	Description	Threat
.scr	Windows screen saver file	Can contain worms and Trojans
.bat	DOS batch file	Can contain malicious commands
.pif	Program information file	Can run malicious programs
.com	DOS application	Can be a malicious program
.exe	Windows application	Can be a malicious program
.vbs	Visual Basic script	Can contain malicious code
.cmd	Command script	Can be used to script malicious batch files
.shs	Shell scrap object	Can mask rogue programs
.vbe	Visual Basic file	Can contain malicious code
.hta	HTML application	Frequently used by worms and Trojans
.reg	Windows registry settings file	Modifies Windows registry, can change security settings
.jse	JavaScript encoded file	Can contain malicious code
.wsf	Windows Script File	Can execute malicious code
.sct	Windows Script components file	Can execute malicious code
.wsh	Windows Script Host file	Can execute malicious code
.chm	Microsoft Compiled HTML Help file	Can exploit browser vulnerabilities
.js	JavaScript file	Can contain malicious code
.lnk	Shortcut link	Can be used to automate malicious actions
.cpl	Windows Control Panel file	Can contain malicious code or change security settings
.hlp	Microsoft Help file	Can be used in multiple exploits
.wsc	Windows Shell command file	Can execute malicious code
.shb	Shell scrap object	Can mask rogue programs
.vb	Visual Basic file	Can contain malicious code
.msi	Windows Installer package	Can install malicious programs
.msp	Windows Installer Patch file	Can contain malicious code

Table 4-1 Commonly Blocked File Extensions

File Extension	Description	Threat
.bas	Programs written in the BASIC programming language	Can be malicious code
.crt	Digital certificate	Can be used in exploits to trust malicious code
.ins	Microsoft Internet communication settings	Can change security settings
.isp	Microsoft Internet Service Provider settings	Can change security settings
.msc	Microsoft Management Console settings	Can change security settings
.mst	Windows Installer transform file	Can install malicious code
.ade	Microsoft Access file	Can contain malicious code
.adp	Microsoft Access file	Can contain malicious code
.mdb	Microsoft Access file	Can contain malicious code
.inf	Microsoft software and driver installation package	Can install malicious code

Table 4-1 Commonly Blocked File Extensions (*continued*)

As large as Table 4-1 is, many readers can probably add other file extensions to the list from their own experience. Only you can judge what file extensions have an acceptable cost/benefit ratio and should be allowed into your network. However, allowing every file extension into your network is asking for a security exploit. For example, Visual Basic script (.vbs) files are one of the most common malicious file types for e-mail worms and viruses. Although people rarely send each other .vbs files for legitimate reasons, worms and viruses do it all the time. It only makes sense to block .vbs files from automatically entering your network.

Dangerous file extensions can be blocked at the Internet gateway device, e-mail server, or e-mail client. A plethora of commercial and open source programs exist to block file attachments at the gateway and e-mail server level. In addition, most antivirus vendors offer an e-mail server antivirus solution.

Blocking Outlook File Attachments Many administrators believe that they cannot block potentially dangerous file extensions in their network. They believe end users and management would revolt. But when management hears the statistics, they present a compelling business argument for file blocking. According to the Radicati Group in April 2010, there were at that time 294 billion e-mails sent each day globally on the Internet. That's 2.8 million e-mails per second, and 90 trillion per year. Of those, 90% contain spam and viruses. This means that spam and viruses comprise:

- 2,520,000 e-mails per second
- 264,600,000,000 e-mails per day
- 81,000,000,000,000 e-mails per year

Even if you have a spam filtering service, which you should (or you'll be overwhelmed by all the spam), some malicious e-mails will slip through. If necessary, you can compromise by allowing blocked file attachments to be sent to a quarantine area where they can be inspected before release. Or you can allow your most savvy users, who can be trusted not to open untrusted files, to have the discretion of turning file blocking off.

E-mail security is essential in today's environment. By preventing malicious HTML content and blocking potentially dangerous file attachments, you have significantly improved the security of your organization.

Install Applications to Nonstandard Directories and Ports

Many malware programs depend on the fact that most people install programs to default directories and on default ports. You can significantly minimize the risk of exploitation by installing programs into nonstandard directories and instructing them to use nonstandard ports. Many Unix and Linux exploits rely on the existence of the /etc directory. By simply changing the installation folder to something other than /etc, you've significantly reduced the risk of malicious attacks being successful. Similarly, instead of installing Microsoft Office to C:\Program Files\Microsoft Office, consider customizing the program during installation to be placed in C:\Program Files\MSOffice. Consider installing Windows into a different folder than the default of C:\Windows. Any change from the default setting, even one character, is enough to defeat many automated attack tools.

If your application opens and uses a TCP/IP port, see if you can make it communicate on a port other than the default. For instance, if you have an extranet web site, consider telling your customers to connect to some other port besides port 80 by using the following syntax in their browser:

```
http://www.domainname.com:X
```

where *X* is the new port number. For example,

```
http://www.mydomain.com:801
```

Many web exploits only check for web servers on port 80, so this change would guard against that attack.

Lock Down Applications

One of the biggest risks to any environment is the ability for an end user to install and run any software they want. There are many tools available to limit what an end user can and cannot run on the desktop. In Windows, the administrator could set system policies to prevent the installation of new applications, take away the user's Run command, and severely limit the desktop. Windows also has a feature called Software Restriction Policies that allows administrators to designate what software is allowed to run on a particular computer. Applications can be defined and allowed by the following methods: trusted digital certificate, hash calculations, placement in an Internet security zone, path location, and file type.

Secure P2P Services

Peer-to-peer (P2P) applications, like instant messaging (IM) and music sharing, are likely to remain strong attack targets in the future. This is because P2P applications have very limited security, if any, and are often installed in the corporate environment without the administrator's authorization. And, they are designed to access files on the end user's computer, which makes the job of stealing those files that much easier. Consequently, P2P applications are seen more as a nuisance than a legitimate service that needs to be secured and managed. However, there are some steps you can take to manage P2P applications and minimize their security consequences.

First, if P2P isn't authorized in your corporate environment, eradicate it. Start by educating end users and working with management to establish penalties for unauthorized software. Then track the programs down and remove them. Tracking them down means monitoring firewall logs for known P2P port attempts, using an IDS on the local network to sniff for P2P packets, or using P2P auditing software.

Second, make sure your firewall is configured to explicitly stop P2P traffic. Because P2P software often uses port 80 as a proxy port, it can be difficult to block P2P traffic by port number alone, but there are things you can do. If the P2P clients connect to servers with a particular IP address or in a particular domain, block the destination at the firewall. Some firewalls allow you to use wildcards in blocked domain names, such as *irc* or *kaz*.

Last, if your end users insist on using P2P, and it is authorized by management, insist on a more secure P2P application, if at all possible. For instance, if your end users insist on using AOL's IM client, see if management will spring for AOL's corporate IM client. It's not free, but it does have more security. There are dozens of secure corporate IM clients available, and all have significantly better security. And finally, make sure the desktop antivirus scanner inspects P2P traffic.

Make Sure Programmers Program Securely

SQL injection and buffer-overflow attacks can only be defeated by programmers using secure coding practices. Type either phrase into an Internet search engine and it will return dozens of documents on how to prevent those types of attacks. Preventing SQL injection attacks can be as simple as using double quotation marks instead of single quotes. Stopping buffer-overflow attacks requires input validation. Several free and commercial tools are available to test your applications for the presence of these attacks and to offer remediation suggestions.

The IIS Lockdown Tool should be executed on any system running IIS. It works by using templates specifically designed for different web server roles (such as OWA server, public web server, and so on). The security templates turn off unnecessary features, remove unneeded files, and install URLScan, which filters out many common, malicious URL attacks. If the installation negatively affects the IIS server, it can easily be uninstalled and the original settings restored.

Back Up the System

With the notable exception of stolen confidential information, the most common symptom of damage from malware is modified, corrupted, or deleted files. Worms and viruses often delete files, format hard drives, or intentionally corrupt data. Even malware that does nothing intentionally wrong to a system's files is maliciously modifying a system just by being present. Security experts cannot always repair the damage and put the system back to the way it was prior to the exploit. This means it's important to keep regular, tested backups of your system. The backup should include all your data files at a minimum, and a complete system backup ensures a quicker recovery in the event of a catastrophic exploit event.

The one caveat to this last piece of advice is to remember that the exploit or hidden malware that damaged your system in the first place could have contaminated your backups and may need to be dealt with prior to putting the system back into production.

Implement ARP Poisoning Defenses

As discussed in Chapter 2, ARP poisoning attacks are one of the most common and effective threats against network infrastructures (especially wireless networks). They are a form of man-in-the-middle (MITM) attack that allows an attacker to intercept and modify network traffic, invisibly. Thus, these attacks merit their own special countermeasures. There are a few ways an organization can defend against an ARP poisoning attack. Defenses include implementing static ARP tables, configuring port rate limiting, or using DHCP snooping with dynamic ARP inspection (DAI). The most effective defense is a combination of the latter two methods

Create a Computer Security Defense Plan

This chapter has covered the steps you can take to secure a computer system. Now you need to take what you've learned and apply it in a comprehensive computer security defense plan. These are the steps to creating a plan:

1. Inventory the assets you have to protect.
2. Decide the value of each asset and its chance of being exploited in order to come up with a quantifiable exposure risk.
3. Using the steps outlined in this chapter (and summarized next), develop a plan to tighten the security on your protected assets. Assets with the highest exposure risk should be given the most protection, but make sure all assets get some baseline level of security.
4. Develop and document security baseline tools and methods. For example, develop an acceptable security template for end-user workstations. Document a method for applying security templates to those workstations (probably a group policy), and put policies and procedures in force to make sure each workstation gets configured with a security template.
5. Use vulnerability testing tools to confirm assets have been appropriately configured.
6. Do periodic testing to make sure security settings stay implemented.
7. Change and update the plan as dictated by new security events and risks.

Although this is a security cliché, security is an ongoing process, not a simple one-time configuration change. Good security means applying reasonable computer security measures consistently and reliably. Good security is boring, but that is the way you want it to be. Exciting computer security, fighting attackers and eradicating computer viruses, means you have holes in your computer security defense.

Implement Static ARP Tables

From a console, if you execute the command arp –a, it will display the ARP table for your system. A quick review of the output shows the IP address and the MAC address associated with the IP address (device). This is how the system knows how to route traffic. One of the devices listed is the gateway address. This is the address for the switch where traffic will pass, if the device wants to send information to a device that doesn't exist in its ARP table. A simple ARP request is sent to ask for the information. The information is then added to the ARP table of the device. The switch follows the same steps to build its ARP table. This is known as dynamic updating and is used for most devices in an organization.

Static ARP tables are exactly what the name implies, static. This means that instead of using the basic ARP request/reply method, the tables are managed by the organization, essentially hard coded. This helps to prevent an ARP poisoning attack because the main avenue of the attack is cut off. The issue with static ARP is the amount of overhead required to keep static ARP tables up to date. If a device doesn't know where to route traffic, essentially the packets will be dropped. This means the user cannot access systems where an entry doesn't exist.

Unfortunately, the payoff for this defense is not worth the effort, which is why organizations don't implement it. Every time a new device is placed on the network, static ARP requires making a manual entry in all other devices in order to properly route traffic through the network. When considering an organization with thousands of employees and devices constantly being changed on the network, this would quickly become an insurmountable task. This solution may be useful in a home environment where systems are rarely replaced.

Configure Port Rate Limiting

Another possible solution for defense is port rate limiting (PRL). In this scenario, the amount of traffic passing over a port during a given length of time is monitored. If the configured threshold is tripped, the port closes itself until either it is enabled manually or a specified length of time passes (usually 15 minutes).

In order to establish an effective threshold, an organization will need to monitor the amount of traffic for a "normal" system over the course of a few weeks. By monitoring traffic correctly, a proper threshold can be set. If the organization does not do its research ahead of time and simply implements what it thinks is a "good" threshold, it may find that its users are constantly exceeding the threshold and unable to perform their day-to-day work. Another possible outcome from this approach is that the threshold will be set too high, which defeats the original purpose for putting PRL in place.

If we look at how an MITM attack with ARP poisoning works, it's easy to see why PRL is a fairly effective defense. As explained earlier, ARP poisoning works by moving the traffic of the victim system(s) through the attacker's device. If an attack is executed on a port with PRL, the amount of traffic should be enough to trip the threshold and thus shut off the port. If the port is unusable to an attacker, you have essentially cut off their ability to perform ARP poisoning from that port. It is essentially a "fail closed" scenario for the organization. If enabling the port requires manual intervention, this could help alert the organization of something suspicious, especially if it happens on several ports within a short timeframe.

PRL requires the attacker to do more research within the organization to set a proper threshold. A motivated attacker will learn from this experience and perhaps perform a more targeted attack in hopes of circumventing this defense. For example, if an attacker is targeting a small group of users, he might execute the attack against a single user at a time. The amount of traffic may not be enough to trip the threshold and the attack may be successful.

Use DHCP Snooping and Dynamic ARP Inspection

The most effective defense against ARP poisoning is to use DHCP snooping with dynamic ARP inspection (DAI). The basis of this defense is that it drops all ARP reply requests not contained within its table. As with PRL, this defense requires the organization to do some research on its environment before full implementation can be executed. The organization needs to run DHCP snooping for two to three weeks in order to build a proper table of IP addresses and MAC addresses. After it has built that table, it can implement DAI. Once implemented, DAI provides a solid defense against ARP poisoning attacks.

In this scenario, when the attacker's system tells the switch via ARP reply that his system's MAC address is the victim's MAC address, the switch compares this information with its table and drops the traffic if it doesn't match, thereby cutting off the avenue in which the attacker communicates.

Combine PRL and DAI for the Most Effective Defense

The most effective defense for an organization against ARP poisoning is a combination of port rate limiting and dynamic ARP inspection. This defense-in-depth approach gives the organization the best possibility for preventing an ARP poisoning attack.

The most effective way to prevent ARP poisoning is to replace all network devices with new, attack-resistant devices. This usually requires a substantial financial investment, which is why many organization hesitate to do so, and thus ARP poisoning remains a viable and common attack vector. But without new, secure infrastructure, the organization will continue to be vulnerable and an effective attacker will always be successful.

Summary

This chapter covered the principles that information security practitioners need to know in order to secure technology infrastructures. The CIA triad is perhaps the most well-known model to guide security implementations, with its focus on confidentiality, integrity, and availability of data. However, there are several other models that focus on other aspects of information security that are also important. Those additional aspects should be taken into consideration when designing a security program.

Whether you're talking about a network, a single computer, or any environment from any other branch of security, an onion is always better than a lollipop. The onion represents a layered security strategy, whereas the lollipop represents a single defense. A defense-in-depth strategy is better because it requires attackers to break through many different countermeasures. These security layers can be combined and allocated into different areas of a network, known as zones of trust, based on the criticality, risks, and exposure of the resources located in those zones.

Attacks can come from automated malicious code or from manual assaults by attackers. There are many countermeasures you can implement on computers to minimize the risk of a successful attack, including securing the physical environment to stop direct attacks by attackers who gain physical control of a device, hardening the operating system to reduce the attack surface, keeping patches updated so that vendor-supplied security fixes are applied, using an antivirus scanner to detect and block malware, using firewall software to control who can get in to a computer and what programs can communicate out, securing network shares to stop worms and attackers from spreading malware, using encryption to preserve the confidentiality of data, and securing applications using their built-in security options. Reliable backups are also important, so that systems can always be returned to a known good state. Security settings should be automated whenever possible and should be part of a computer security defense plan.

Finally, ARP poisoning was covered because it's a significant threat in today's networks. Even if all computers are locked down according to best practices, ARP poisoning can be used to take over those computers' network sessions by a Man in the Middle, who would then control all communications. Defenses against ARP poisoning include manual configuration of ARP tables, port rate limiting, and dynamic ARP inspection.

The design of a computer network should take into account all of these principles in order to be as secure as possible. Even so, these are primarily defensive strategies, and as discussed in Chapter 1, detection and deterrence are also required for security. The following chapters provide guidance on that.

References

Andress, Jason, and Steve Winterfeld. *Cyber Warfare: Techniques, Tactics and Tools for Security Practitioners*. Syngress, 2011.

Schneier, Bruce. *Secrets and Lies*. John Wiley & Sons, 2002.

Schwartau, Winn. *Cybershock: Surviving Hackers, Phreakers, Identity Thieves, Internet Terrorists, and Weapons of Mass Disruption*. Thunder's Mouth Press, 2000.

Stoneburner, Gary, Clark Hayden, and Alexis Feringa. *Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Rev. A* (NIST Special Publication 800-27, Rev A). NIST, 2004. http://csrc.nist.gov/publications/nistpubs/800-27A/SP800-27-RevA.pdf.